

**Examenul de bacalaureat național 2014**  
**Proba E. d) – 4 iulie 2014**  
**Informatică**  
**Limbajul C/C++**

**Varianta 4**

**Filiera teoretică, profilul real, specializările: matematică-informatică**  
**matematică-informatică intensiv informatică**  
**Filiera vocațională, profilul militar, specializarea matematică-informatică**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

**I. THEMA** **(30 Puncte)**

Für Punkt 1, schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht.

1. Der Wert des nebenstehenden C/C++ Ausdrucks ist: **(4P.)** | 42/10\*29/10
- a. 6                      b. 8                      c. 11                      d. 18

**2. Sei der nebenstehende Pseudocode-Algorithmus.**

Man bezeichnet mit  $x \% y$  den Rest der Teilung der natürlichen Zahl  $x$  durch die natürliche, von Null verschiedene Zahl  $y$  und mit  $[z]$  den ganzen Teil der reellen Zahl  $z$ .

- a) Schreibt die angeschriebenen Werte wenn die Zahl 2352 eingelesen wird. **(6P.)**
- b) Schreibt zwei Zahlen mit höchstens zwei Ziffern die eingelesen werden können, so dass nach dem Durchführen des Algorithmus, für jede diese, die Werte 5 1 angeschrieben werden. **(4P.)**
- c) Schreibt in Pseudocode einen, mit dem gegeben, äquivalenten Algorithmus, in dem man die erste **solange...wiederhole** Struktur mit einer anderen Wiederholungsstruktur ersetzen soll. **(6P.)**

```
lies n
(natürliche, von Null verschiedene Zahl)
d ← 2
solange d ≤ n wiederhole
    p ← 0
    solange n % d = 0 wiederhole
        p ← p + 1
        n ← [n / d]
    wenn p % 2 = 0 und p ≠ 0 dann
        schreibe d, ' '
    d ← d + 1
schreibe n
```

- d) Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(10P.)**

## II. THEMA

(30 Puncte)

Für jeden der Punkte 1 und 2 schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht.

1. Ein gerichteter Graph hat 8 Spitzen, beschriftet von 1 bis 8 und die Bögen (1,7), (1,8), (3,5), (3,7), (4,3), (4,7), (6,3), (6,5), (6,7), (6,8), (8,5), (8,7). Die Anzahl der Spitzen die den äußeren Grad Null haben, sind: (4P.)

a. 1                      b. 2                      c. 3                      d. 4

2. Die Variable `s` kann eine Folge von maximal 20 Zeichen speichern. Nach dem Durchführen der nebenstehenden Anweisungssequenz wird folgendes angeschrieben: (4P.)

```
strcpy(s, "1b2d3");  
s[2]='a'+2;  
strcpy(s, s+1);  
strcpy(s+3, s+4);  
cout<<s;      |      printf("%s", s);
```

a. 1b438                      b. 1bcd8                      c. ba2                      d. bcd

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

3. Sei nebenstehende Deklaration. Schreibt eine Anweisungssequenz nach deren Durchführung auf dem Bildschirm die die Nachricht `acceptat` angeschrieben wird, wenn der Zeitmoment entsprechend der Variablen `start` dem Zeitmoment `aus derselben Stunde` entsprechend der Variablen `stop` vorgängig ist, oder die Nachricht `respins` im Gegenfall. (6P.)
- ```
struct timp  
{    int minut;  
   int secunda;  
} start, stop;
```
4. Man nimmt an das die Höhe eines Baumes mit Wurzel gleich mit der größten der Längen der elementaren Ketten, die eine Extremität in der Wurzel des Baumes und die andere Extremität in jedwelcher der "Blätter" des Baumes hat, ist.  
Sei ein Baum mit 9 Knoten, beschriftet von 1 bis 9 und die Kanten [1,2], [2,3], [2,5], [3,7], [4,5], [5,6], [5,8], [8,9]. Schreibt die Knoten die als Wurzel gewählt werden können, so dass die Höhe des Baumes maximal sein soll. (6P.)
5. Schreibt ein C/C++ Programm das von der Tastatur zwei natürliche Zahlen `m` und `n` ( $3 \leq m \leq 50$ ,  $3 \leq n \leq 50$ ) und die Elemente eines zweidimensionalen Feldes mit `m` Reihen und `n` Spalten, natürliche höchstens vierstellige Zahlen einliest und nachher im Speicher das Feld verändert, indem es die vorletzte Reihe und die vorletzte Spalte dieses löscht, wie im Beispiel. Das Programm schreibt auf dem Bildschirm das so erhaltene Feld, jede Reihe des Feldes auf je einer Bildschirmzeile, die Elemente derselben Reihe sind getrennt durch je ein Leerzeichen.

**Beispiel:** für `m=4`, `n=5` und das Feld

```
5 1 2 3 4  
8 2 2 5 3  
2 1 7 3 9  
3 0 9 8 5
```

wird auf dem Bildschirm, das unterstehenden Feld angeschrieben:

```
5 1 2 4  
8 2 2 3  
3 0 9 5
```

(10P.)

**III. THEMA**

**(30 Puncte)**

**Für Punkt 1, schreibt auf das Prüfungsblatt den Buchstaben welcher der richtigen Antwort entspricht.**

1. Sei das Unterprogramm **f**, nebenstehend definiert. Gebt den Wert von **f(15)** an. (4P.)
- ```
int f(int n)
{ if (n<10) return f(n+1)+3;
  else if (n==10) return 7;
    else return f(n-2)-1;
}
```
- a. 1                      b. 7                      c. 8                      d. 10

**Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.**

2. Die Backtracking Methode benützend, erzeugt alle Möglichkeiten der Form Ketten von je 4 Perlen von unterschiedlichen Farben aus der Menge {roșu, galben, roz, albastru, violet}, so dass in jeder Kette auf nebenstehenden Positionen keine roten (roșu) und gelben (galben) Perlen sein können. Zwei Ketten sind unterschiedlich wenn sie wenigstens eine Perle von unterschiedlicher Farbe haben oder wenn die Reihenfolge der Farbe der Perlen unterschiedlich ist.  
Die ersten fünf erzeugten Lösungen, in dieser Reihenfolge, sind: (roșu, roz, galben, albastru), (roșu, roz, galben, violet), (roșu, roz, albastru, galben), (roșu, roz, albastru, violet), (roșu, roz, violet, galben). Schreibt die sechste und die siebente Lösung, in der Reihenfolge in der sie erzeugt wurden. (6P.)
3. Ein Intervall mit der Eigenschaft das es eine einzige natürliche Zahl, **n** ( $2 \leq n$ ) gibt, für die der Wert des Produktes  $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$  diesem Intervall gehört, wird **interval factorial** von **n** genannt.  
**Beispiel:** [5, 8] și [3, 23] sind intervale factoriale von 3, aber [1, 15] und [7, 10] sind keine intervale factoriale für keine Zahl.  
Sei das Unterprogramm **interval**, mit drei Parameter:
- **n**, durch den es eine natürliche Zahl aus dem Intervall [2, 10] bekommt.
  - **a** und **b**, durch die es je eine natürliche Zahl liefert ( $a < b$ ) mit der Eigenschaft dass der Ausdruck **b-a** maximalen Wert haben soll und [a, b] ist ein interval factorial von **n**.
- Schreibt die vollständige Definition des Unterprogramms.  
**Beispiel:** wenn **n=3**, nach dem Aufruf **a=3** und **b=23**. (10P.)
4. Eine natürliche Zahl **x**, gebildet aus genau zwei Ziffern, wird **sub-număr** einer natürlichen Zahl **y** genannt, wenn die Ziffern von **x** in derselben Reihenfolge, auf aufeinanderfolgenden Positionen in der Zahl **y** erscheinen.  
**Beispiel:** 21 ist sub-număr von 12145, von 213, von 21, aber nicht von 123 oder von 231.  
Die Datei **bac.txt** enthält höchstens 1000000 natürliche Zahlen aus dem Intervall [10, 10<sup>9</sup>], getrennt durch je ein Leerzeichen.  
Schreibt auf dem Bildschirm, getrennt durch je ein Leerzeichen, die sub-numere die am meisten in der Schreibweise der Zahlen aus der Datei erscheinen. Für das Bestimmen der verlangten sub-numere wird einer in Bezug auf die Laufzeit effizienter Algorithmus benützt.  
**Beispiel:** wenn die Datei **bac.txt** die Zahlen  
393 17775787 72194942 12121774  
enthält, dann werden auf dem Bildschirm die unterstehenden Werte angeschrieben, nicht unbedingt in dieser Reihenfolge:  
77 21  
a) Beschreibt in der Umgangssprache den Algorithmus und begründet seine Effizienz. (4P.)  
b) Schreibt das dem beschriebenen Algorithmus entsprechende C/C++ Programm. (6P.)